Closed sets based discovery of small covers for association rules

Nicolas Pasquier, Yves Bastide, Rafik Taouil, Lotfi Lakhal

Closed Set Based Discovery of Small Covers for Association Rules

Nicolas Pasquier Yves Bastide Rafik Taouil Lotfi Lakhal

Laboratoire d'Informatique (LIMOS)
Université Blaise Pascal - Clermont-Ferrand II
Complexe Scientifique des Cézeaux
24 Avenue des Landais, 63177 Aubière Cedex France
{pasquier,bastide,taouil,lakhal}@libd2.univ-bpclermont.fr

Abstract

In this paper, we address the problem of the usefulness of the set of discovered association rules. This problem is important since real-life databases yield most of the time several thousands of rules with high confidence. We propose new algorithms based on Galois closed sets to reduce the extraction to small covers, or bases, for exact and approximate rules. Once frequent closed itemsets – which constitute a generating set for both frequent itemsets and association rules – have been discovered, no additional database pass is needed to derive these bases. Experiments conducted on real-life databases show that these algorithms are efficient and valuable in practice.

Keywords: data mining, Galois closure operator, frequent closed itemsets, bases for association rules, algorithms.

1 Introduction and Motivation

Data mining has been extensively addressed for the last years, specially the problem of discovering association rules. The aim when discovering association rules is to exhibit relationships between data items (or attributes) and compute the precision of each relationship in the database. Usual precision measures are support and confidence [1] that point the proportion of database transactions (or objects) upholding each rule out. When an association rule has support and confidence exceeding some user-defined minimum thresholds, the rule is considered as relevant and the extracted knowledge would likely be used for supporting decision making. A classical example of association rules fits in the context of market basket data analysis and highlights a particular feature in customers behavior: 80% of customers who buy cereals and sugar also buy milk.

Since the problem was stated [1], various approaches have been proposed for an increased efficiency of rule discovery [2, 4, 8, 17, 23, 24, 26, 30, 33]. However, fully taking advantage of exhibited knowledge means capabilities to handle such a knowledge. In fact, by using a synthetic dataset containing 100,000 objects, each of which encompassing around 10 items, our experiments yield more than 16,000 rules with confidence outcoming 90%. The problem is much more critical when collected data is highly correlated or dense, like in statistical or medical databases. For instance, when applied to a census dataset of 10,000 objects, each of which characterized by values of 73 attributes, experiments result in more than 2,000,000 rules with support and confidence outcoming 90%.

Thus the talked issue could be rephrased as follows: which relevant knowledge can be learned from several thousands of rules highly redundant? Which aid could be offered to users for handling countless rules and focusing on useful ones? Before explaining how our approach answers the previous questions, let us examine proposed solutions for meeting such needs.

1.1 Related Work: an Outline

Among approaches addressing the described issue, two main trends can be distinguished. The former provides users with mechanisms for filtering rules. In [3, 16], the user defines templates, and rules not matching with them are discarded. In [22, 29], boolean operators are introduced for selecting rules including (or not) given items. A similar approach expanded with a measure of usefulness of extracted rules, called improvement, is proposed in [5]. In [21], an SQL-like operator called `MINE RULE`, allowing the specification of general extraction criteria, is proposed. The quoted approaches operate "a posteriori", i.e. once huge amount of rules are extracted, querying facilities make it possible to handle rule subsets selected according to the user preferences. In contrast, the second trend addresses the problem with an "a priori" vision, by attempting to minimize the number of exhibited rules. In [14, 28], information about taxonomies are used to define criteria of interest which apply for pruning redundant rules. In [7, 25], statistical measures such as Pearson's correlation or the chi-squared test are used instead of the confidence measure.

1.2 Contribution: an Overview

The approach presented in this paper belongs to the second trend since it aims to extract not all possible rules but a sub-set called small cover or basis for association rules. When computing such a basis, redundant rules are discarded since they do not vehicule relevant knowledge. Such a pruning operation is a key-step during rule extraction, and significantly reduces the resulting set. For example, experiments performed using a real-life dataset describing characteristics of mushrooms yield the 9 following association rules with *free gills* in the antecedent and *eatable* in the consequent, and with common support (51%) and confidence (54%).

1) *free gills* $\rightarrow$ *eatable*
2) *free gills* $\rightarrow$ *eatable, partial veil*
3) *free gills* $\rightarrow$ *eatable, white veil*
4) *free gills* $\rightarrow$ *eatable, partial veil, white veil*
5) *free gills, white veil* $\rightarrow$ *eatable*
6) *free gills, white veil* $\rightarrow$ *eatable, partial veil*
7) *free gills, partial veil* $\rightarrow$ *eatable*
8) *free gills, partial veil* $\rightarrow$ *eatable, white veil*
9) *free gills, white veil, partial veil* $\rightarrow$ *eatable*

Among these rules, 8 are redundant because they can be deduced from the 4^{th} rule: *free gills* $\rightarrow$ *eatable, partial veil, white veil.* Moreover, since rules unexpected by the user are important [18, 27], presenting a list of rules covering all the frequent items in the dataset is also needed.

First, using the closure operator of the Galois connection [6], we characterize frequent closed itemsets [23, 24]. Then, we show that frequent closed itemsets represent a generating set for both frequent itemsets and association rules. The underlying theorem states the foundations of our approach since it makes it possible to generate the bases from frequent closed itemsets by avoiding handling of large sets of rules. We propose two new algorithms: the former achieves frequent closed itemsets from frequent itemsets without accessing the dataset, and the latter, called Apriori-Close, extends the Apriori algorithm [2] by discovering simultaneously frequent itemsets and frequent closed itemsets without additional execution time.

Then, using the frequent closed itemsets and the pseudo-closed itemsets defined by Duquenne and Guigues in lattice theory [9, 11], we define the *Duquenne-Guigues basis for exact association rules* (rules with a 100% confidence). Rules in this basis are non-redundant exact rules with minimal antecedent and maximal consequent. Besides, using the frequent closed itemsets and results proposed by Luxenburger in lattice theory [19, 32], we define the *proper basis* and the *structural basis for approximate association rules.* The proper basis is a small set containing the most informative and useful approximate rules: the non-redundant informative rules. The structural basis can be viewed as an abstract of all approximate rules that hold and can be useful when the proper basis is large. We propose three algorithms intended for yielding these three bases. Using the set of frequent closed itemsets, generating the evoked bases is performed without any access to the dataset.

An algorithm discovering closed and pseudo-closed itemsets has been proposed in [12] and implemented in ConImp [9]. However, this algorithm does not consider the support of itemsets and, since it works

only in main memory, it cannot be applied when the number of objects exceeds some hundreds and the number of items some tens. From the results presented in [19], no algorithm was proposed. In [24], the association rule framework based on the Galois connection is defined. Fitting in this groundwork, two efficient algorithms that discover frequent closed itemsets for association rules are defined: the Close algorithm [24] for correlated data and the A-Close algorithm [23] for weakly correlated data. The work presented in this paper differs from [23, 24] in the following points:

1. It shows that frequent closed itemsets constitute a generating set for frequent itemsets and association rules.
2. It extends the Apriori algorithm and algorithms for discovering maximal frequent itemsets to generate frequent closed itemsets.
3. It adapts the Duquenne-Guigues basis and Luxenburger results for exact and partial implications to the context of association rules. This adaptation is based on 1. (generating set).
4. It presents new algorithms for generating bases for exact and approximate association rules using frequent closed itemsets.
5. It shows that the algorithms proposed are efficient for both improving the usefulness of extracted association rules and decreasing the execution time of the association rule extraction.

As shown by experiments, the proposed process for extracting bases does not require any overhead compared with the traditional approaches for discovering association rules.

1.3 Paper Organization

In Section 2, we present the association rule framework based on the Galois connection. Section 3 addresses the concept of basis for both exact and approximate association rules. New algorithms for discovering frequent and frequent closed itemsets are described in Section 4 and the following section presents algorithms computing the bases for association rules from the frequent closed itemsets. Experimental results achieved from various datasets are given in Section 6. Finally, as a conclusion, we evoke further work in Section 7.

2 Association Rule Framework

In this section, we present the association rule framework based on the Galois connection, primarily introduced in [23, 24].

Definition 1 (Data mining context) *A data mining context*[1] *is defined as* $\mathcal{D} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$, *where* $\mathcal{O}$ *and* $\mathcal{I}$ *are finite sets of objects and items respectively.* $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$ *is a binary relation between objects and items. Each couple* $(o, i) \in \mathcal{R}$ *denotes the fact that the object* $o \in \mathcal{O}$ *is related to the item* $i \in \mathcal{I}$.

Depending on the target system, a data mining context can be a relation, a class, or the result of an SQL/OQL query.

Example 1 An example data mining context $\mathcal{D}$ consisting of 5 objects (identified by their OID) and 5 items is illustrated in Table 1.

Definition 2 (Galois connection) *Let* $\mathcal{D} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ *be a data mining context. For* $O \subseteq \mathcal{O}$ *and* $I \subseteq \mathcal{I}$, *we define:*

$$f: 2^{\mathcal{O}} \rightarrow 2^{\mathcal{I}} \qquad f(O) = \{i \in \mathcal{I} \mid \forall o \in O, (o, i) \in \mathcal{R}\}$$

$$g: 2^{\mathcal{I}} \rightarrow 2^{\mathcal{O}} \qquad g(I) = \{o \in \mathcal{O} \mid \forall i \in I, (o, i) \in \mathcal{R}\}$$

[1] By extension, we will call dataset a data mining context.

OID	Items			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		
5	A	B	C	E

Table 1: The Example Data Mining Context $\mathcal{D}$.

$f(O)$ associates with O the items common to all objects $o \in O$ and $g(I)$ associates with I the objects related to all items $i \in I$. The couple of applications (f, g) is a Galois connection between the power set of $\mathcal{O}$ ($2^{\mathcal{O}}$) and the power set of $\mathcal{I}$ ($2^{\mathcal{I}}$). The following properties hold for all $I, I_1, I_2 \subseteq \mathcal{I}$ and $O, O_1, O_2 \subseteq \mathcal{O}$:

$$(1)\ \ I_1 \subseteq I_2 \Rightarrow g(I_1) \supseteq g(I_2) \qquad (1')\ O_1 \subseteq O_2 \Rightarrow f(O_1) \supseteq f(O_2)$$
$$(2)\ \ O \subseteq g(I) \Longleftrightarrow I \subseteq f(O)$$

Definition 3 (Frequent itemsets) *Let $I \subseteq \mathcal{I}$ be a set of items from $\mathcal{D}$. The support count of the itemset I in $\mathcal{D}$ is:*

$$supp(I) = \frac{\|g(I)\|}{\|\mathcal{O}\|}$$

I is said to be frequent if the support of I in $\mathcal{D}$ is at least minsupp. The set L of frequent itemsets in $\mathcal{D}$ is:

$$L = \{I \subseteq \mathcal{I} \mid supp(I) \geq minsupp\}$$

Definition 4 (Association rules) *An association rule is an implication between two itemsets, with the form $I_1 \rightarrow I_2$ where $I_1, I_2 \subseteq \mathcal{I}$, $I_1, I_2 \neq \varnothing$ and $I_1 \cap I_2 = \varnothing$. I_1 and I_2 are called respectively the antecedent and the consequent of the rule. The support supp(r) and confidence conf(r) of an association rule $r : I_1 \rightarrow I_2$ are defined using the Galois connection as follows:*

$$supp(r) = \frac{\|g(I_1 \cup I_2)\|}{\|\mathcal{O}\|}, \qquad conf(r) = \frac{supp(I_1 \cup I_2)}{supp(I_1)}$$

Association rules holding in the context are those that have support and confidence greater than or equal to the minsupp and minconf thresholds respectively. We define the set AR of association rules holding in $\mathcal{D}$ given minsupp and minconf thresholds as follows:

$$AR = \{r : I_1 \rightarrow I_2 - I_1 \mid I_1 \subset I_2 \subseteq \mathcal{I} \wedge supp(I_2) \geq minsupp \wedge conf(r) \geq minconf\}$$

If conf(r)=1 then r is called an exact association rule or implication rule, otherwise r is called approximate association rule.

Example 2 Exact and approximate association rules extracted from $\mathcal{D}$ for *minsupp* = 2/5 and *minconf* = 1/2 are given in Table 2.

3 Bases for Association Rules

In this section, we first demonstrate that the frequent closed itemsets constitute a generating set for frequent itemsets and association rules. Then, we characterize the *Duquenne-Guigues basis for exact association rules* and the *proper* and *structural bases for approximate association rules*. The Duquenne-Guigues basis, as defined in [11], is extended in this paper to the context of association rules. Proofs of Theorems 2, 3 and 4 are straightforward from Theorem 1 and [11, 19, 32]. Interested readers could refer to [6, 31] for further details on closed sets.

Exact rule	Supp	Approximate rule	Supp	Conf	Approximate rule	Supp	Conf
ABC ⇒ E	2/5	BCE → A	2/5	2/3	B → AE	2/5	2/4
ABE ⇒ C	2/5	AC → BE	2/5	2/3	E → AB	2/5	2/4
ACE ⇒ B	2/5	BE → AC	2/5	2/4	A → CE	2/5	2/3
AB ⇒ CE	2/5	CE → AB	2/5	2/3	C → AE	2/5	2/4
AE ⇒ BC	2/5	AC → B	2/5	2/3	E → AC	2/5	2/4
AB ⇒ C	2/5	BC → A	2/5	2/3	B → CE	3/5	3/4
AB ⇒ E	2/5	BE → A	2/5	2/4	C → BE	3/5	3/4
AE ⇒ B	2/5	AC → E	2/5	2/3	E → BC	3/5	3/4
AE ⇒ C	2/5	CE → A	2/5	2/3	A → B	2/5	2/3
BC ⇒ E	3/5	BE → C	3/5	3/4	B → A	2/5	2/4
CE ⇒ B	3/5	A → BCE	2/5	2/3	C → A	3/5	3/4
A ⇒ C	3/5	B → ACE	2/5	2/4	A → E	2/5	2/3
B ⇒ E	4/5	C → ABE	2/5	2/4	E → A	2/5	2/4
E ⇒ B	4/5	E → ABC	2/5	2/4	B → C	3/5	3/4
		A → BC	2/5	2/3	C → B	3/5	3/4
		B → AC	2/5	2/4	C → E	3/5	3/4
		C → AB	2/5	2/4	E → C	3/5	3/4
		A → BE	2/5	2/3			

Table 2: Association Rules Extracted from $\mathcal{D}$ for $minsup = 2/5$ and $minconf = 1/2$.

3.1 Generating Set

Definition 5 (Galois closure operators) *The operators $h = f \circ g$ in $2^{\mathcal{I}}$ and $h' = g \circ f$ in $2^{\mathcal{O}}$ are Galois closure operators[2]. Given the Galois connection (f, g), the following properties hold for all $I, I_1, I_2 \subseteq \mathcal{I}$ and $O, O_1, O_2 \subseteq \mathcal{O}$ [6]:*

Extension :	*(3)* $I \subseteq h(I)$	*(3')* $O \subseteq h'(O)$
Idempotency :	*(4)* $h(h(I)) = h(I)$	*(4')* $h'(h'(O)) = h'(O)$
Monotonicity :	*(5)* $I_1 \subseteq I_2 \Rightarrow h(I_1) \subseteq h(I_2)$	*(5')* $O_1 \subseteq O_2 \Rightarrow h'(O_1) \subseteq h'(O_2)$

Definition 6 (Frequent closed itemsets) *An itemset $I \subseteq \mathcal{I}$ in $\mathcal{D}$ is a closed itemset iff $h(I) = I$. A closed itemset I is said to be frequent if the support of I in $\mathcal{D}$ is at least minsupp. The smallest (minimal) closed itemset containing an itemset I is $h(I)$, the closure of I. The set FC of frequent closed itemsets in $\mathcal{D}$ is defined as follows:*

$$FC = \{I \subseteq \mathcal{I} \mid I = h(I) \ \wedge \ supp(I) \geq minsupp\}$$

Example 3 A frequent closed itemset is a maximal set of items common to a set of objects, for which support is at least *minsupp*. The frequent closed itemsets in the context $\mathcal{D}$ for *minsupp*=2/5 are presented in Table 3. The itemset BCE is a frequent closed itemset since it is the maximal set of items common to the objects $\{2, 3, 5\}$. The itemset BC is not a frequent closed itemset since it is not a maximal set of items common to some objects: all objects in relation with the items B and C (objects 2, 3 and 5) are also in relation with the item E.

Hereafter, we demonstrate that the set of frequent closed itemsets with their support is the smallest collection from which frequent itemsets with their support and association rules can be generated (it is a generating set).

Lemma 1 *[24] The support of an itemset I is equal to the support of the smallest closed itemset containing I: $supp(I) = supp(h(I))$.*

Lemma 2 *[24] The set of maximal frequent itemsets $M = \{I \in L \mid \nexists\, I' \in L \text{ where } I \subset I'\}$ is identical to the set of maximal frequent closed itemsets $MC = \{I \in FC \mid \nexists\, I' \in FC \text{ where } I \subset I'\}$.*

[2]Here, we use the following notation: $f \circ g(I) = f(g(I))$ and $g \circ f(O) = g(f(O))$.

Frequent closed itemset	Support
{∅}	5/5
{C}	4/5
{AC}	3/5
{BE}	4/5
{BCE}	3/5
{ABCE}	2/5

Table 3: Frequent Closed Itemsets Extracted from $\mathcal{D}$ for *minsupp* = 2/5.

Theorem 1 (Generating set) *The set FC of frequent closed itemsets with their support is a generating set for all frequent itemsets and their support, and for all association rules holding in the dataset, their support and their confidence.*

Proof. Based on Lemma 2, all frequent itemsets can be derived from the maximal frequent closed itemsets. Based on Lemma 1, the support of each frequent itemset can be derived from the support of frequent closed itemsets. Then, the set of frequent closed itemsets FC is a generating set for both the set of frequent itemsets L and the set of association rules AR[3]. □

3.2 Duquenne-Guigues Basis for Exact Association Rules

Definition 7 (Frequent pseudo-closed itemsets) *An itemset $I \subseteq \mathcal{I}$ in $\mathcal{D}$ is a pseudo-closed itemset iff $h(I) \neq I$ and $\forall I' \subset I$ such as I' is a pseudo-closed itemset, we have $h(I') \subset I$. The set FP of frequent pseudo-closed itemsets in $\mathcal{D}$ is defined as*

$$FP = \{I \subseteq \mathcal{I} \mid supp(I) \geq minsupp \wedge I \neq h(I) \wedge \forall I' \in FP \text{ such as } I' \subset I \text{ we have } h(I') \subset I\}$$

Theorem 2 (Duquenne-Guigues Basis for Exact Association Rules) *Let FP be the set of frequent pseudo-closed itemsets in $\mathcal{D}$. The set*

$$DG = \{r : I_1 \Rightarrow h(I_1) - I_1 \mid I_1 \in FP \wedge I_1 \neq \varnothing\}$$

is a basis for all exact association rules holding in the dataset.

The Duquenne-Guigues basis is minimal with respect to the number of rules since there can be no complete set with fewer rules than there are frequent pseudo-closed itemsets [10, 13].

Example 4 A frequent pseudo-closed itemset I is a frequent non-closed itemset that includes the closures of all frequent pseudo-closed itemsets included in I. The set FP of frequent pseudo-closed itemsets and the Duquenne-Guigues basis for exact association rules extracted from $\mathcal{D}$ for *minsupp*=2/5 and *minconf*=1/2 are presented in Table 4. The itemset AB is not a frequent pseudo-closed itemset since the closures of A and B (respectively AC and BE) are not included in AB. $ABCE$ is not a frequent pseudo-closed itemset since it is closed.

Frequent pseudo-closed itemset	Support
{A}	3/5
{B}	4/5
{E}	4/5

Exact rule	Support
A ⇒ C	3/5
B ⇒ E	4/5
E ⇒ B	4/5

Table 4: Frequent Pseudo-Closed Itemsets and Duquenne-Guigues Basis Extracted from $\mathcal{D}$ for *minsupp* = 2/5.

[3] Furthermore, FC is the smallest generating set for L and AR. Hence, even if frequent itemsets can be derived from the maximal frequent itemsets, passes over the dataset are still needed to compute the frequent itemset supports.

3.3 Proper Basis for Approximate Association Rules

Theorem 3 (Proper Basis for Approximate Association Rules) *Let FC be the set of frequent closed itemsets in $\mathcal{D}$. The set*

$$PB = \{r : I_1 \rightarrow I_2 - I_1 \mid I_1, I_2 \in FC \wedge I_1 \neq \varnothing \wedge I_1 \subset I_2 \wedge conf(r) \geq minconf\}$$

is a basis for all approximate association rules holding in the dataset. Association rules in PB are proper approximate association rules.

Example 5 The proper basis for approximate association rules extracted from $\mathcal{D}$ for *minsupp*=2/5 and *minconf*=1/2 are presented in Table 5.

Approximate rule	Support	Confidence
BCE $\rightarrow$ A	2/5	2/3
AC $\rightarrow$ BE	2/5	2/3
BE $\rightarrow$ AC	2/5	2/4
BE $\rightarrow$ C	3/5	3/4
C $\rightarrow$ ABE	2/5	2/4
C $\rightarrow$ BE	3/5	3/4
C $\rightarrow$ A	3/5	3/4

Table 5: Proper Basis Extracted from $\mathcal{D}$ for $minsupp = 2/5$ and $minconf = 1/2$.

3.4 Structural Basis for Approximate Association Rules

Definition 8 (Undirected graph $\mathcal{G}_{FC}$) *Let FC be the set of frequent closed itemsets in $\mathcal{D}$. We define $\mathcal{G}_{FC} = (V, E)$ as the undirected graph associated with FC where the set of vertices V and the set of edges E are defined as follows:*

$$V = \{I \subseteq \mathcal{I} \mid I \in FC\}$$

$$E = \{(I_1, I_2) \in V \times V \mid I_1 \subset I_2 \wedge supp(I_2)/supp(I_1) \geq minconf\}$$

With each edge in $\mathcal{G}_{FC}$ between two vertices I_1 and I_2 with $I_1 \subset I_2$ is associated the confidence = $supp(I_2)$ / $supp(I_1)$ of the proper approximate association rule $I_1 \rightarrow I_2 - I_1$ represented by the edge.

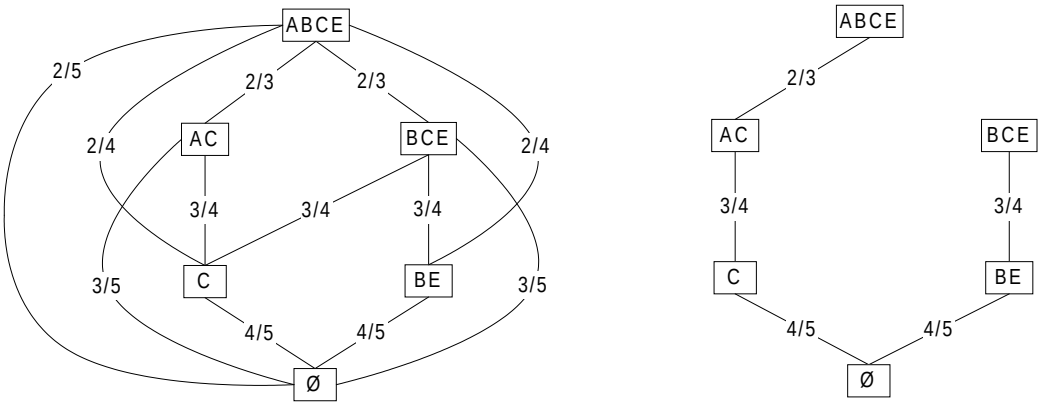
Definition 9 (Maximal Confidence Spanning Forest $\mathcal{F}_{FC}$) *Let $\mathcal{F}_{FC} = (V, E')$ be the maximal confidence spanning forest associated with FC. $\mathcal{F}_{FC}$ is obtained from the undirected graph $\mathcal{G}_{FC} = (V, E)$ by suppressing transitive edges and cycles. Cycles are removed by deleting some edges that enter the last vertex I (maximal vertex with respect to the inclusion) of the cycle. Among all edges entering in I, those with confidence less than the maximal confidence value associated with an edge with the form $(I', I) \in E$ are deleted. If more than one edge have the maximal confidence value, the first one in lexicographic order is kept.*

Theorem 4 (Structural Basis for Approximate Association Rules) *Let SB be the set of association rules represented by edges in $\mathcal{F}_{FC}$ except rules from the vertex $\{\varnothing\}$. The set*

$$SB = \{r : I_1 \rightarrow I_2 - I_1 \mid I_1, I_2 \in V \wedge I_1 \subset I_2 \wedge I_1 \neq \varnothing \wedge (I_1, I_2) \in E'\}$$

is a basis for all approximate association rules holding in the dataset ($\mathcal{I}$ is the consequent of at most one approximate association rule in SB).

Example 6 The structural basis for approximate association rules extracted from $\mathcal{D}$ for *minsupp*=2/5 and *minconf*=1/2 is presented in Table 6.



$\mathcal{G}_{FC}$ $\mathcal{F}_{FC}$

Figure 1: Undirected Graph $\mathcal{G}_{FC}$ and Maximal Confidence Spanning Forest $\mathcal{F}_{FC}$ (a tree in this example) Derived from $\mathcal{D}$ for $minsupp = 2/5$ and $minconf = 1/2$.

Approximate rule	Support	Confidence
AC $\rightarrow$ BE	2/5	2/3
BE $\rightarrow$ C	3/5	3/4
C $\rightarrow$ A	3/5	3/4

Table 6: Structural Basis Extracted from $\mathcal{D}$ for $minsupp = 2/5$ and $minconf = 1/2$.

4 Discovering Frequent and Frequent Closed Itemsets

In Section 4.1, we propose a new algorithm to achieve frequent closed itemsets from frequent itemsets without accessing the dataset. This algorithm discovers frequent closed itemsets while for instance an algorithm for discovering maximal frequent itemsets [4, 17, 33] is used. In Section 4.2, we present an extension of the Apriori algorithm [2] called Apriori-Close for discovering frequent and frequent closed itemsets without additional computation time. Like in the Apriori algorithm, we assume in the following that items are sorted in lexicographic order and that k is the size of the largest frequent itemsets. Based on Lemma 2, k is also the size of the largest frequent closed itemsets.

4.1 Computing Frequent Closed Itemsets from Frequent Itemsets

Many efficient algorithms for mining frequent itemsets and their support have been proposed. Well-known proposals are presented in [2, 8, 26, 30]. Efficient algorithms for discovering the maximal frequent itemsets and then achieve all frequent itemsets have also been proposed [4, 17, 33]. All these algorithms give as result the set $L = \bigcup_{i=1}^{i=k} L_i$ where L_i contains all frequent i-itemsets (itemsets of size i). Based on Proposition 1 and Lemma 2 (Section 3.1), the frequent closed itemsets and their support can be computed from the frequent itemsets and their support without any dataset access.

The pseudo-code to determine frequent closed itemsets among frequent itemsets is given in Algorithm 1. Notations are given in Table 7. The input of the algorithm are sets L_i, $1 \leq i \leq k$, containing all frequent itemsets in the dataset. It recursively generates the sets FC_i, $0 \leq i \leq k$, of frequent closed i-itemsets from FC_k to FC_0.

L_i	Set of frequent i-itemsets and their support.
FC_i	Set of frequent closed i-itemsets and their support.
isclosed	Variable indicating if the considered itemset is closed or not.

Table 7: Notations.

Proposition 1 *The support of a closed itemset is greater than the supports of all its supersets.*

Proof. Let l be a closed i-itemset and s a superset of l. We have $l \subset s \Rightarrow g(l) \supseteq g(s)$ (Property *(1)* of the Galois connection). If $g(l) = g(s)$ then $h(l) = h(s) \Rightarrow l = h(s) \Rightarrow s \subseteq l$ (absurd). It follows that $g(l) \supset g(s) \Rightarrow supp(l) > supp(s)$. □

Algorithm 1 Deriving Frequent Closed Itemsets from Frequent Itemsets.

```
1)  FC_k ← L_k;
2)  for (i ← k−1; i ≠ 0; i--) do begin
3)      FC_i ← {};
4)      forall itemsets l ∈ L_i do begin
5)          isclosed ← true;
6)          forall itemsets l' ∈ L_{i+1} do begin
7)              if (l ⊂ l') and (l.support = l'.support) then isclosed ← false;
8)          end
9)          if (isclosed = true) then FC_i ← FC_i ∪ {l};
10)     end
11) end
12) FC_0 ← {∅};
13) forall itemsets l ∈ L_1 do begin
14)     if (l.support = ||O||) then FC_0 ← {};
15) end
```

First, the set FC_k is initialized with the set of largest frequent itemsets L_k (step 1). Then, the algorithm iteratively determines which i-itemsets in L_i are closed from L_{k-1} to L_1 (steps 2 to 11). At the beginning of the i^{th} iteration the set FC_i of frequent closed i-itemsets is empty (step 3). In steps 4 to 10, for each frequent itemset l in L_i, we verify that l has the same support as a frequent $(i+1)$-itemset l' in L_{i+1} in which it is included. If so, we have $l' \subseteq h(l)$ and then $l \neq h(l)$: l is not closed (step 7). Otherwise, l is a frequent closed itemset and is inserted in FC_i (step 9). During the last phase, the algorithm determines if the empty itemset is closed by first initializing FC_0 with the empty itemset (step 12) and then considering all frequent 1-itemsets in L_1 (steps 13 to 15). If a 1-itemset l has a support equal to the number of objects in the context, meaning that l is common to all objects, then the itemset $\varnothing$ cannot be closed (we have $supp(\{\varnothing\}) = \|\mathcal{O}\| = supp(l)$) and is removed from FC_0 (step 14). Thus, at the end of the algorithm, each set FC_i contains all frequent closed i-itemsets.

Correctness Since all maximal frequent itemsets are maximal frequent closed itemsets (Lemma 2), the computation of the set FC_k containing the largest frequent closed itemsets is correct. The correctness of the computation of sets FC_i for $i<k$ relies on Proposition 1. This proposition enables to determine if a frequent i-itemset l is closed by comparing its support and the supports of the frequent $(i+1)$-itemsets in which l is included. If one of them has the same support as l, then l cannot be closed.

4.2 Apriori-Close Algorithm

In this section, we present an extension of the Apriori algorithm [2] computing simultaneously frequent and frequent closed itemsets. The pseudo-code is given in Algorithm 2 and notations in Table 8. The algorithm iteratively generates the sets L_i of frequent i-itemsets from L_1 to L_k. Besides, during the i^{th} iteration, all frequent closed $(i-1)$-itemsets in FC_{i-1} are determined. The set FC_k is determined during the last step of the algorithm.

L_i	Set of frequent i-itemsets, their support and marker *isclosed* indicating if closed or not.
FC_i	Set of frequent closed i-itemsets and their support.

Table 8: Notations.

First, the variable k is initialized to 0 (step 1). Then, the set L_1 of frequent 1-itemsets is initialized with the list of items in the context (step 2) and one pass is performed to compute their support (step

Algorithm 2 Discovering Frequent and Frequent Closed Itemsets with Apriori-Close.

```
1)  k ← 0;
2)  itemsets in L_1 ← {1-itemsets};
3)  L_1 ← Support-Count(L_1);
4)  FC_0 ← {∅};
5)  forall itemsets l ∈ L_1 do begin
6)      if (l.support < minsupp) then L_1 ← L_1 \ {l};
7)      else if (l.support = ||O||) then FC_0 ← {};
8)  end
9)  for (i ← 1; L_i ≠ {}; i++) do begin
10)     forall itemsets l' ∈ L_i do l'.isclosed ← true;
11)     L_{i+1} ← Apriori-Gen(L_i);
12)     forall itemsets l ∈ L_{i+1} do begin
13)         forall i-subsets l' of l do begin
14)             if (l' ∉ L_i) then L_{i+1} ← L_{i+1} \ {l};
15)         end
16)     end
17)     L_{i+1} ← Support-Count(L_{i+1});
18)     forall itemsets l ∈ L_{i+1} do begin
19)         if (l.support < minsupp) then L_{i+1} ← L_{i+1} \ {l};
20)         else do begin
21)             forall i-subsets l' ∈ L_i of l do begin
22)                 if (l.support = l'.support) then l'.isclosed ← false;
23)             end
24)         end
25)     end
26)     FC_i ← {l ∈ L_i | l.isclosed = true};
27)     k ← i;
28) end
29) FC_k ← L_k;
```

3). The set FC_0 is initialized with the empty itemset (step 4) and the supports of itemsets in L_1 are considered (steps 5 to 8). All infrequent 1-itemsets are removed from L_1 (step 6) and if a frequent 1-itemset has a support equal to the number of objects in the context then the empty itemset is removed from FC_0 (step 7). During each of the following iterations (steps 9 to 28), frequent itemsets of size i+1, $k > i \geq 1$, and frequent closed itemsets of size i are computed as follows. For all frequent i-itemsets in L_i, the marker *isclosed* is initialized to *true* (step 10). A set L_{i+1} of possible frequent (i+1)-itemsets is created by applying the Apriori-Gen function to the set L_i (step 11). For each of these possible frequent (i+1)-itemsets, we check that all its subsets of size i exist in L_i (steps 12 to 16). One pass is performed to compute the supports of the remaining itemsets in L_{i+1} (step 17). Then, for each (i+1)-itemsets $l \in L_{i+1}$ (steps 18 to 25), if l is infrequent then it is discarded from L_{1+1} (step 19). Otherwise for all i-subsets l' of l, we verify that supports of l' and l are equal; if so, then l' cannot be a closed itemset and its marker *isclosed* is set to false (steps 20 to 24). Then, all frequent i-itemsets in L_i for which marker *isclosed* is *true* are inserted in the set FC_i of frequent closed i-itemsets (step 26) and the variable k is set to the value of i (step 27). Finally, the set FC_k is initialized with the frequent k-itemsets in L_k (step 29).

Apriori-Gen function The Apriori-Gen function [2] applies to a set L_i of frequent i-itemsets. It returns a set L_{i+1} of potential frequent (i+1)-itemsets. A new itemset in L_{i+1} is created by joining two itemsets in L_i sharing common first i-1 items.

Support-Count function The Support-Count function takes a set L_i of i-itemsets as argument. It efficiently computes the supports of all itemsets $l \in L_i$. Only one dataset pass is required: for each object o read, the supports of all itemsets $l \in L_i$ that are included in the set of items associated with o, i.e. $l \subseteq f(\{o\})$, are incremented. The subsets of $f(\{o\})$ are quickly found using the Subset function described

in Section 5.2.

Correctness Since the support of a frequent closed itemset l is different from the support of all its supersets (Proposition 1), the computation of sets FC_i for $i < k$ is correct. Hence, a frequent i-itemset $l' \in L_i$ is determined closed or not by comparing its support with the supports of all frequent $(i+1)$-itemsets $l \in L_{i+1}$ for which $l' \subset l$. Lemma 2 ensures the correctness of the computation of the set FC_k containing the largest frequent closed itemsets.

Example 7 Figure 2 illustrates the execution of the Apriori-Close algorithm with the context $\mathcal{D}$ for a minimum support of 2/5.

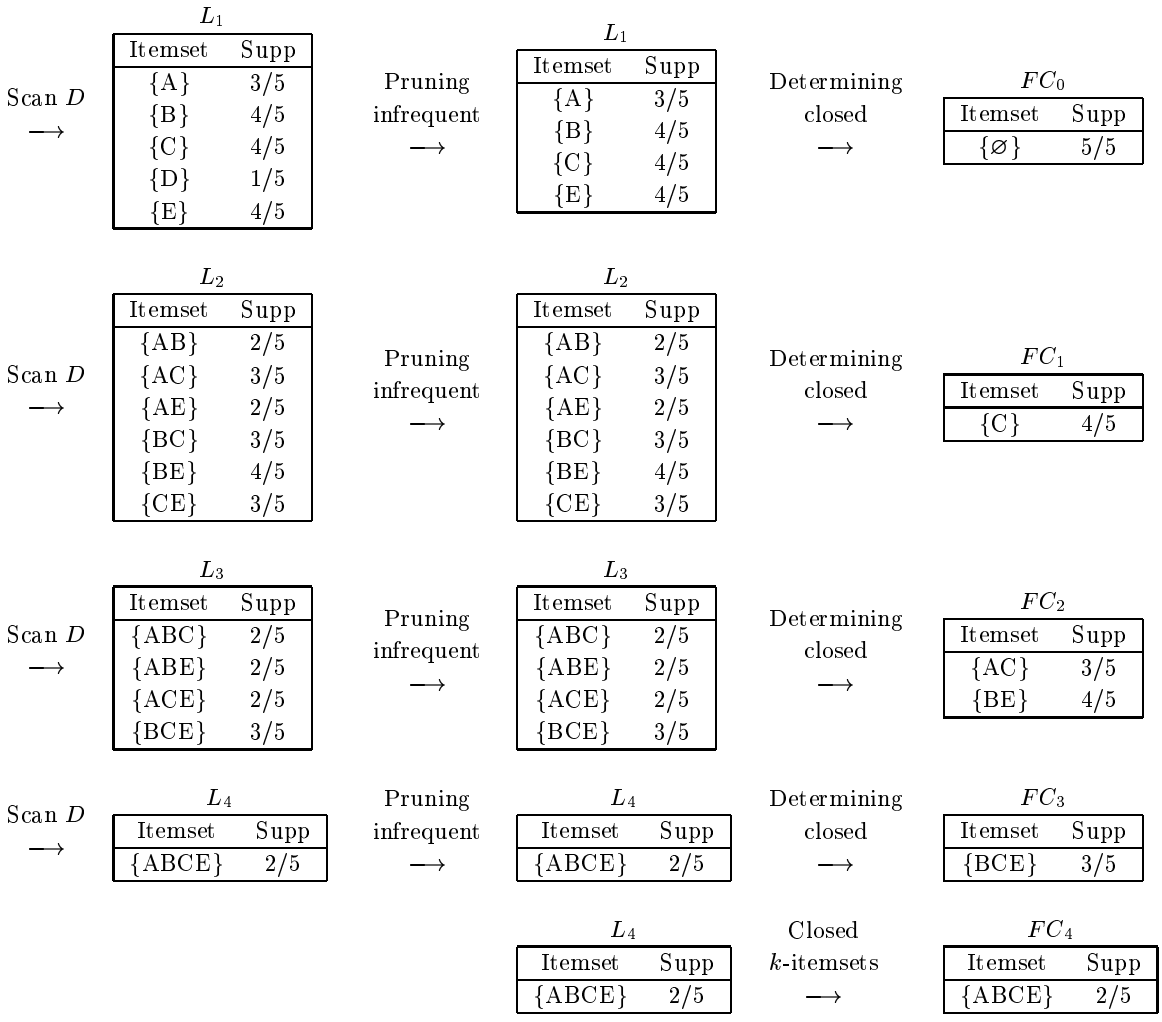


Figure 2: Discovering Frequent and Frequent Closed Itemsets with Apriori-Close.

5 Generating Bases for Association Rules

In Section 5.1, we present an algorithm to generate the Duquenne-Guigues basis for exact association rules. In Sections 5.2 and 5.3 are described algorithms achieving the proper basis and the structural basis for approximate association rules respectively.

5.1 Generating Duquenne-Guigues Basis for Exact Association Rules

The pseudo-code generating the Duquenne-Guigues basis for exact association rules is given in Algorithm 3. Notations are given in Table 9. The algorithm takes as input the sets L_i, $1 \leq i \leq k$, containing the frequent itemsets and their support, and the sets FC_i, $0 \leq i \leq k$, containing the frequent closed itemsets and their support. It first computes the frequent pseudo-closed itemsets iteratively (steps 2 to 17) and then uses them to generate the Duquenne-Guigues basis for exact association rules DG (steps 18 to 22).

L_i	Set of frequent i-itemsets and their support.
FC_i	Set of frequent closed i-itemsets and their support.
FP_i	Set of frequent pseudo-closed i-itemsets, their closure and their support.
DG	Duquenne-Guigues basis for exact association rules.

Table 9: Notations.

First, the set DG is initialized to the empty set (step 1). If the empty itemset is not a closed itemset (it is then necessarily a pseudo-closed itemset), it is inserted in FP_0 (step 2). Otherwise FP_0 is empty (step 3). Then, the algorithm recursively determines which i-itemsets in L_i are pseudo-closed from L_1 to L_k (steps 4 to 16). At each iteration, the set FP_i is initialized with the list of frequent i-itemsets that are not closed (step 5) and each frequent i-itemsets l in FP_i is considered as follows (steps 6 to 15). The variable *pseudo* is set to *true* (step 7). We verify for each frequent pseudo-closed itemset p previously discovered (i.e. in FP_j with $j < i$) if p is contained in l (steps 8 to 13). In that case and if the closure of p is not included in l, then l is not pseudo-closed and is removed from FP_i (steps 9 to 12). Otherwise, the closure of l (i.e. the smallest frequent closed itemset containing l) is determined (step 14). Once all frequent pseudo-closed itemsets p and their closure are computed, all rules with the form $r : p \Rightarrow (p.\text{closure} - p)$ are generated (steps 17 to 21). The algorithm results in the set DG containing all rules in the Duquenne-Guigues basis for exact association rules.

Algorithm 3 Generating Duquenne-Guigues Basis for Exact Association Rules.

1) $DG \leftarrow \{\}$;
2) **if** $(FC_0 = \{\})$ **then** $FP_0 \leftarrow \{\varnothing\}$;
3) **else** $FP_0 \leftarrow \{\}$;
4) **for** $(i \leftarrow 1;\ i \leq k;\ i{+}{+})$ **do begin**
5) $\quad FP_i \leftarrow L_i \setminus FC_i$;
6) $\quad$ **forall** itemsets $l \in FP_i$ **do begin**
7) $\quad\quad pseudo \leftarrow true$;
8) $\quad\quad$ **forall** itemsets $p \in FP_j$ with $j < i$ **do begin**
9) $\quad\quad\quad$ **if** $(p \subset l)$ **and** $(p.\text{closure} \not\subset l)$ **then do begin**
10) $\quad\quad\quad\quad pseudo \leftarrow false$;
11) $\quad\quad\quad\quad FP_i \leftarrow FP_i \setminus \{l\}$;
12) $\quad\quad\quad$ **end**
13) $\quad\quad$ **end**
14) $\quad\quad$ **if** $(pseudo = true)$ **then** $l.\text{closure} \leftarrow Min_{\subseteq}(\{c \in FC_{j>i} \mid l \subseteq c\})$;
15) $\quad$ **end**
16) **end**
17) **forall** sets FP_i where $FP_i \neq \{\}$ **do begin**
18) $\quad$ **forall** pseudo-closed itemsets $p \in FP_i$ **do begin**
19) $\quad\quad DG \leftarrow DG \cup \{r : p \Rightarrow (p.\text{closure}{-}p), p.\text{support}\}$;
20) $\quad$ **end**
21) **end**

Correctness Since the itemset $\varnothing$ has no subset, if it is not a closed itemset then it is by definition a pseudo-closed itemset and the computation of the set FP_0 is correct. The correctness of the computation of frequent pseudo-closed i-itemsets in FP_i for $1 \leq i \leq k$ relies on Definition 7. All frequent i-itemsets l

in L_i that are not closed, i.e. not in FC_i, are considered. Those l containing the closures of all frequent pseudo-closed itemsets that are subsets of l are inserted in FP_i. According to Definition 7, these i-itemsets are all frequent pseudo-closed i-itemsets and the sets FP_i are correct. The association rules generated in the last phase of the algorithm are all rules with a frequent pseudo-closed itemset in the antecedent. Then, the resulting set DG corresponds to the rules in the Duquenne-Guigues basis for exact association rules defined in Theorem 2.

Example 8 Figure 3 shows the generation of the Duquenne-Guigues basis for exact association rules from the context $\mathcal{D}$ for a minimum support of 2/5.

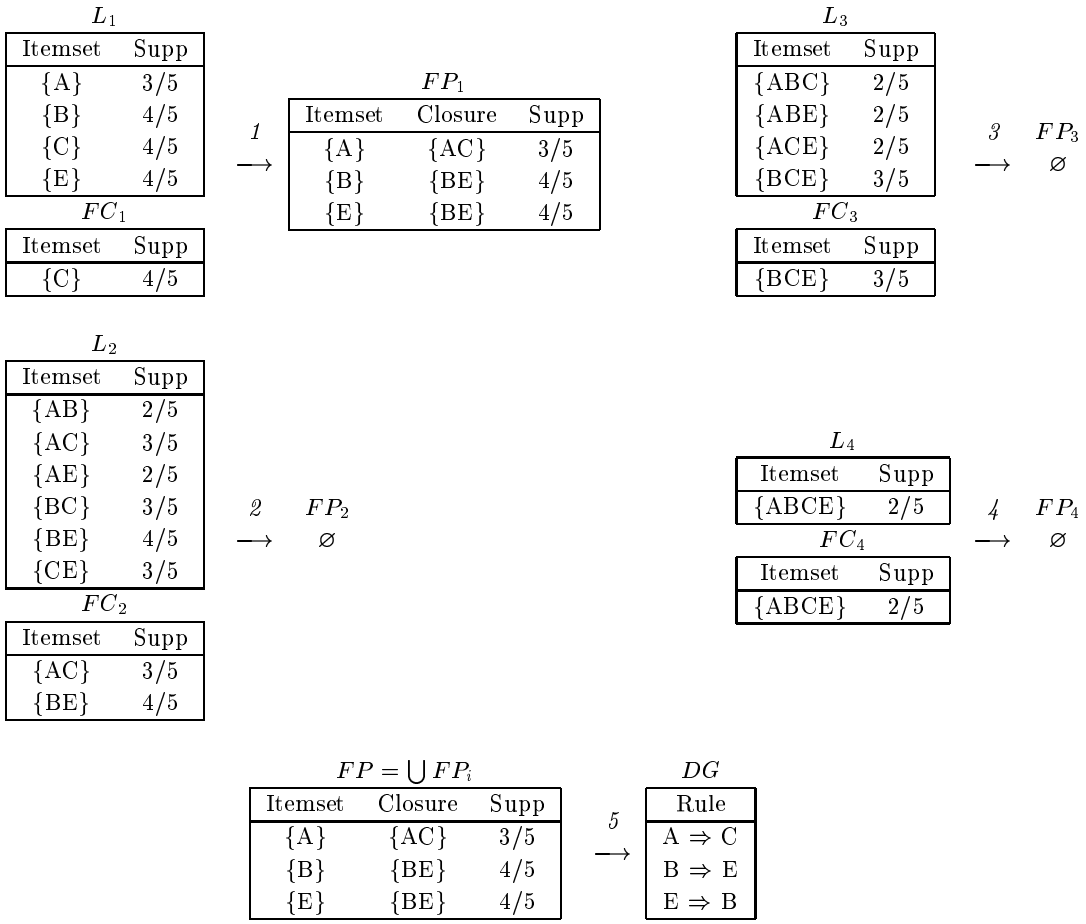


Figure 3: Generating Duquenne-Guigues Basis for Exact Association Rules.

5.2 Generating Proper Basis for Approximate Association Rules

The pseudo-code generating the proper basis for approximate association rules is presented in Algorithm 4. Notations are given in Table 10. The algorithm takes as input the sets FC_i, $1 \leq i \leq k$, containing the frequent closed non-empty itemsets and their support. The output of the algorithm is the proper basis for approximate association rules PB.

The set PB is first initialized to the empty set (step 1). Then, the algorithm iteratively considers all frequent closed itemsets $l \in FC_i$ for $2 \leq i \leq k$. It determines which frequent closed itemsets $l' \in FC_{j<i}$ are subsets of l and generates association rules with the form $l' \rightarrow l - l'$ that have sufficient confidence (steps 2 to 12) as follows. During the i^{th} iteration, each itemset l in FC_i is considered (steps 3 to 11). For each set FC_j, $1 \leq j < i$, a set S_j containing all frequent closed j-itemsets in FC_j that are subsets of l is created (step 5). Then, for each of these subsets $l' \in S_j$ (steps 6 to 9), we compute the confidence of

FC_i	Set of frequent closed i-itemsets and their support.
S_j	Set of j-itemsets that are subsets of the considered itemset.
PB	Proper basis for approximate association rules.

Table 10: Notations.

the proper approximate association rule $r : l' \rightarrow l - l'$ (step 7). If the confidence of r is sufficient then r is inserted in PB (step 8). At the end of the algorithm, the set PB contains all rules of the proper basis for approximate association rules.

Algorithm 4 Generating Proper Basis for Approximate Association Rules.

```
1)  PB ← {}
2)  for (i ← 2; i ≤ k; i++) do begin
3)      forall itemsets l ∈ FC_i do begin
4)          for (j ← i−1; j > 0; j--) do begin
5)              S_j ← Subsets(FC_j, l);
6)              forall itemsets l' ∈ S_j do begin
7)                  conf(r) ← l.support / l'.support;
8)                  if (conf(r) ≥ minconf) then PB ← PB ∪ {r : l' → l − l', l.support, conf(r)};
9)              end
10)         end
11)     end
12) end
```

Subset function The subset function takes a set X of itemsets and an itemset y as arguments. It determines all itemsets $x \in X$ that are subsets of y. In algorithm implementation, frequent and frequent closed itemsets are stored in a *prefix-tree* structure [24] in order to improve efficiency of the subset search.

Correctness The correctness of the algorithm relies on the fact that we inspect all proper approximate association rules holding in the dataset. For each frequent closed itemset, the algorithm computes, among its subsets, all other frequent closed itemsets. Then, the generation of all rules between two frequent closed itemsets having sufficient confidence is ensured. These rules are all proper approximate association rules holding in the dataset, and the resulting set PB is the proper basis for approximate association rules defined in Theorem 3.

Example 9 Figure 4 shows the generation of the proper basis for approximate association rules in the context $\mathcal{D}$ for a minimum support of 2/5 and a minimum confidence of 1/2.

5.3 Generating Structural Basis for Approximate Association Rules

The pseudo-code generating the structural basis for approximate association rules is given in Algorithm 5. Notations are given in Table 11. The algorithm takes as input the sets FC_i, $1 \leq i \leq k$, of frequent closed non-empty itemsets and their support. It generates the structural basis for approximate association rules SB represented by the maximal confidence spanning forest $\mathcal{F}_{FC}$ associated with $FC = \bigcup_{i=1}^{i=k} FC_i$ (without the empty itemset).

FC_i	Set of frequent closed i-itemsets and their support.
S_j	Set of j-itemsets that are subsets of the itemset considered.
CR	Set of candidate approximate association rules.
SB	Structural basis for approximate association rules.

Table 11: Notations.

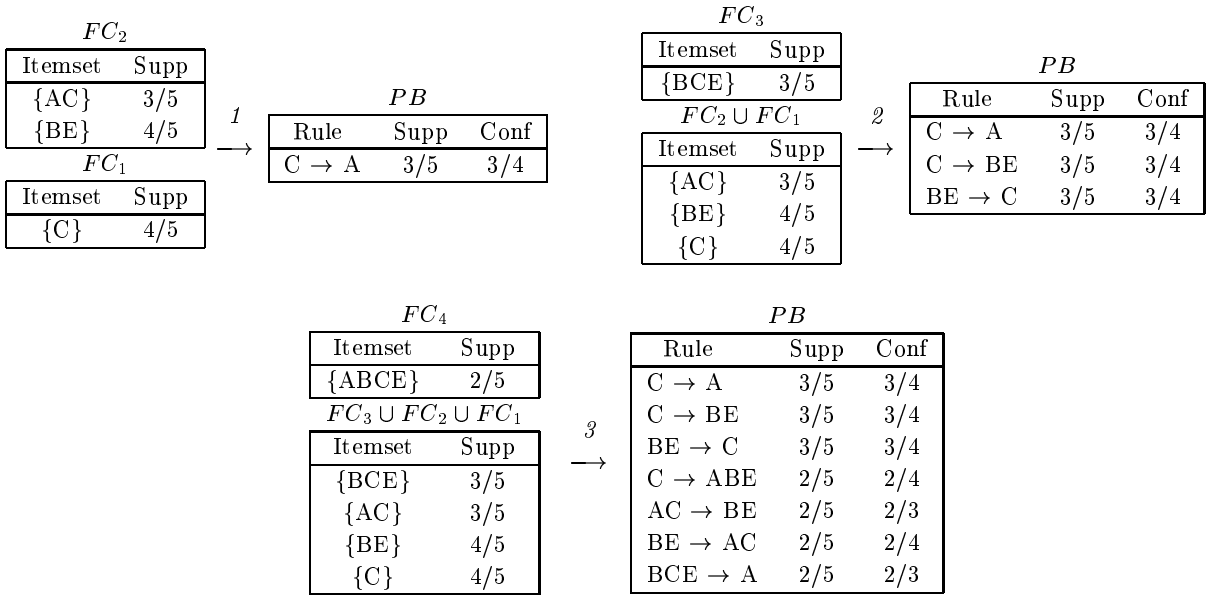


Figure 4: Generating Proper Basis for Approximate Association Rules.

The set SB is first initialized to the empty set (step 1). Then, the algorithm iteratively considers all frequent closed itemsets $l \in FC_i$ for $2 \leq i \leq k$. It determines which frequent closed itemsets $l' \in FC_{j<i}$ are covered by l, i.e. are direct predecessors of l, and then generates the maximal confidence association rules with the form $l \rightarrow l' - l$ that hold (steps 2 to 25). During the i^{th} iteration, each itemset l in FC_i is considered (steps 3 to 24) as follows. The set CR of candidate association rules with l in the consequent is initialized to the empty set (step 4). For $1 \leq j < i$, sets S_j containing all frequent closed j-itemsets in FC_j that are subsets of l are created (steps 5 to 7). Then, all these subsets of l are considered in decreasing order of their sizes (steps 8 to 18). For each of these subsets $l' \in S_j$, the confidence of the proper approximate association rule $r : l' \rightarrow l - l'$ is computed (step 10). If the confidence of r is sufficient, r is inserted in CR (step 12) and all subsets l'' of l' are removed from $S_{n<j}$ (steps 13 to 15). This because rules with the form $l'' \rightarrow l - l''$ with $l'' \in S_{n<j}$ are transitive proper approximate rules. Finally, the candidate proper approximate rules with l in the consequent that are in CR are pruned (steps 19 to 23): the maximum confidence value *maxconf* of rules in CR is determined (step 20) and the first rule with such a confidence is inserted in SB (steps 21 and 22). At the end of the algorithm, the set SB thus contains all rules in the structural basis for approximate association rules.

Correctness The algorithm considers all association rules $l' \rightarrow l - l'$ with confidence $\geq$ *minconf* between two frequent closed itemsets l and l' where l covers l'. These rules are all proper non-transitive approximate association rules that hold and can be represented by the edges of the graph $\mathcal{G}_{FC}$ (Definition 8) without transitive edges. Moreover, among all rules with the form $X \rightarrow l - X$ (generated from l), we keep only the first one with confidence equal to the maximal confidence of rules $X \rightarrow l - X$. Only preserving this rule is equivalent to the cycle removing in the graph $\mathcal{G}_{FC}$ in the same manner as explained in Definition 9. Then, the resulting set SB can be represented as the maximal confidence spanning forest $\mathcal{F}_{FC}$ without edges from the empty itemset. SB contains all rules in the structural basis for approximate association rules defined in Theorem 4.

Example 10 Figure 5 depicts the generation of the structural basis for approximate association rules in the context $\mathcal{D}$ for a minimum support of 2/5 and a minimum confidence of 1/2.

Algorithm 5 Generating Structural Basis for Approximate Association Rules.

```
1)  SB ← {};
2)  for (i ← 2; i ≤ k; i++) do begin
3)      forall itemsets l ∈ FC_i do begin
4)          CR ← {};
5)          for (j ← i−1; j > 0; j--) do begin
6)              S_j ← Subsets(FC_j, l);
7)          end
8)          for (j ← i−1; j > 0; j--) do begin
9)              forall itemsets l' ∈ S_j do begin
10)                 conf(r) ← l.support / l'.support;
11)                 if (conf(r) ≥ minconf) then do begin
12)                     CR ← CR ∪ {r : l' → l − l', l.support, conf(r)};
13)                     for (n ← j−1; n > 0; n--) do begin
14)                         S_n ← S_n − Subsets(S_n, l');
15)                     end
16)                 end
17)             end
18)         end
19)         if (CR ≠ {}) then do begin
20)             maxconf ← Max_{r∈CR}(conf(r));
21)             find first {r ∈ CR | conf(r) = maxconf};
22)             SB ← SB ∪ {r};
23)         end
24)     end
25) end
```

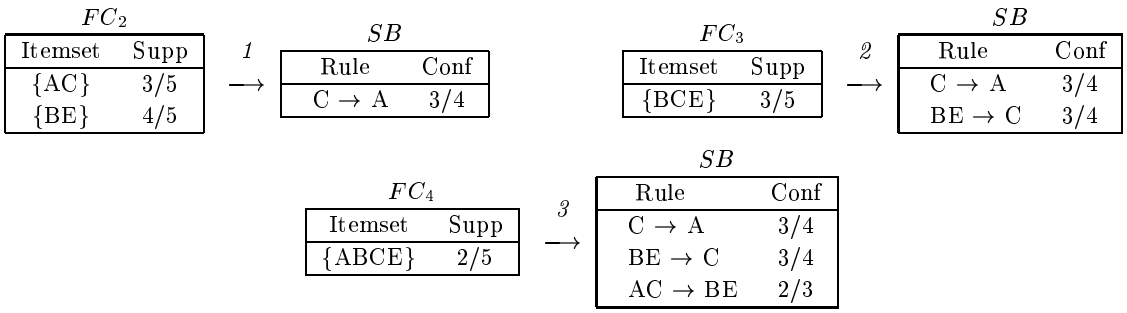


Figure 5: Generating Structural Basis for Approximate Association Rules.

6 Experimental Results

Experiments were performed on a Pentium II PC with a 350 Mhz clock rate, 128 MBytes of RAM, running the Linux operating system. Algorithms were implemented in C++. Characteristics of the datasets used are given in Table 12. These datasets are the T10I4D100K[4] synthetic dataset that mimics market basket data, the C20D10K and the C73D10K census datasets from the PUMS sample file[5] , and the MUSHROOMS[6] dataset describing mushroom characteristics. In all experiments, we attempted to choose significant minimum support and confidence threshold values: we observed threshold values used in other papers for experiments on similar data types and inspected rules extracted in the bases.

[4] http://www.almaden.ibm.com/cs/quest/syndata.html
[5] ftp://ftp2.cc.ukans.edu/pub/ippbr/census/pums/pums90ks.zip
[6] ftp://ftp.ics.uci.edu/~cmerz/mldb.tar.Z

Name	Number of objects	Average size of objects	Number of items
T10I4D100K	100,000	10	1,000
MUSHROOMS	8,416	23	127
C20D10K	10,000	20	386
C73D10K	10,000	73	2,177

Table 12: Datasets.

6.1 Relative Performance of Apriori and Apriori-Close

We conducted experiments to compare response times obtained with Apriori and Apriori-Close on the four datasets. Results for the T10I4D100K and MUSHROOMS datasets are presented in Table 13. We can observe that execution times are identical for the two algorithms: adding the frequent closed itemset derivation to the frequent itemset discovery does not induce additional computation time. Similar results were obtained for C20D10K and C73D10K datasets.

Minsupp	Apriori	Apriori-Close
2.0%	1.99s	1.97s
1.0%	3.47s	3.46s
0.5%	9.62s	9.70s
0.25%	15.02s	14.92s

T10I4D100K

Minsupp	Apriori	Apriori-Close
90%	0.28s	0.28s
70%	0.73s	0.73s
50%	2.40s	2.70s
30%	18.22s	17.93s

MUSHROOMS

Table 13: Execution Times of Apriori and Apriori-Close.

6.2 Number of Rules and Execution Times of the Rule Generation

Table 14 shows the total number of exact association rules and their number in the Duquenne-Guigues basis for exact rules. Table 15 shows the total number of approximate association rules, their number in the proper basis and in the structural basis for approximate rules, and the number of non-transitive rules in the proper basis for approximate rules (5^{th} column). For example in the context $\mathcal{D}$, rules $C \rightarrow A$ and $AC \rightarrow BE$ are extracted, as well as the rule $C \rightarrow ABE$ which is clearly transitive. Since by construction, its confidence – retrieved by multiplying the confidences of the two former – is less than theirs, this rule is the less interesting among the three. Reducing the extraction to non-transitive rules in the proper basis for approximate rules can also be interesting. Such rules are generated by a variant of Algorithm 5 with the last pruning strategy (steps 20 and 21) removed: all candidate rules in CR are inserted in SB.

Table 16 shows for the four datasets the average relative size of bases compared with the sets of all rules obtained. In the case of weakly correlated data (T10I4D100K), no exact rule is generated and the proper basis for approximate rules contains all approximate rules that hold. The reason is that, in such data, all frequent itemsets are frequent closed itemsets. In the case of correlated data (MUSHROOMS, C20D10K and C73D10K), the number of extracted rules in bases is much smaller than the total number of rules that hold.

Figure 6 shows for each dataset the execution times of the computation of all rules (using the algorithm described in [2]) and bases. Execution times of the derivation of the Duquenne-Guigues basis for exact rules and the proper basis for non-transitive approximate rules are not presented since they are identical to those of the derivation of the Duquenne-Guigues basis for exact rules and the structural basis for approximate rules (*Duquenne-Guigues and structural bases*).

7 Conclusion

In this paper, we present new algorithms for efficiently generating bases for association rules. A basis is a set of non-redundant rules from which all association rules can be derived, thus it captures all useful

Dataset	Minsupp	Exact rules	Duquenne-Guigues basis
T10I4D100K	0.5%	0	0
MUSHROOMS	30%	7,476	69
C20D10K	50%	2,277	11
C73D10K	90%	52,035	15

Table 14: Number of Exact Association Rules Extracted.

Dataset (Minsupp)	Minconf	Approximate rules	Proper basis	Non-transitive basis	Structural basis
T10I4D100K (0.5%)	90%	16,260	16,260	3,511	916
	70%	20,419	20,419	4,004	1,058
	50%	21,686	21,686	4,191	1,140
	30%	22,952	22,952	4,519	1,367
MUSHROOMS (30%)	90%	12,911	806	563	313
	70%	37,671	2,454	968	384
	50%	56,703	3,870	1,169	410
	30%	71,412	5,727	1,260	424
C20D10K (50%)	90%	36,012	4,008	1,379	443
	70%	89,601	10,005	1,948	455
	50%	116,791	13,179	1,948	455
	30%	116,791	13,179	1,948	455
C73D10K (90%)	95%	1,606,726	23,084	4,052	939
	90%	2,053,896	32,644	4,089	941
	85%	2,053,936	32,646	4,089	941
	80%	2,053,936	32,646	4,089	941

Table 15: Number of Approximate Association Rules Extracted.

Dataset	Duquenne-Guigues basis	Proper basis	Non-transitive basis	Structural basis
T10I4D100K	-	100.00%	20.05%	5.49%
MUSHROOMS	0.92%	6.90%	2.69%	1.19%
C20D10K	0.48%	11.21%	2.33%	0.63%
C73D10K	0.03%	1.55%	0.21%	0.05%

Table 16: Average Relative Size of Bases.

information. Moreover, its size is significantly reduced compared with the set of all possible rules because redundant, and thus useless, rules are discarded. Our approach has a twofold advantage: on one hand, the user is provided with a smaller set of resulting rules, easier to handle, and vehiculing information of improved quality. On the other hand, execution times are reduced compared with the discovering of all association rules. Such results are proved (in the groundwork of lattice theory) and illustrated by experiments, achieved from real-life datasets.

Integrating reduction methods Templates, as defined in [3, 16], can directly be used for extracting from the bases all association rules matching some user specified patterns. Information in taxonomies associated with the dataset can also be integrated in the process as proposed in [14, 28] for extracting bases for generalized (multi-level) association rules. Integrating item constraints and statistical measures, such as described in [5, 22, 29] and [7, 25] respectively, in the generation of bases requires further work.

Functional and approximate dependencies Algorithms presented in this paper can be adapted to generate bases for functional and approximate dependencies. In [15, 20], such bases and algorithms for generating them were proposed. However, the Duquenne-Guigues basis is smaller than the basis for functional dependencies constituted of minimal non-trivial functional dependencies. Hence, the number of rules in the Duquenne-Guigues basis is minimal; moreover these rules have minimal antecedent and

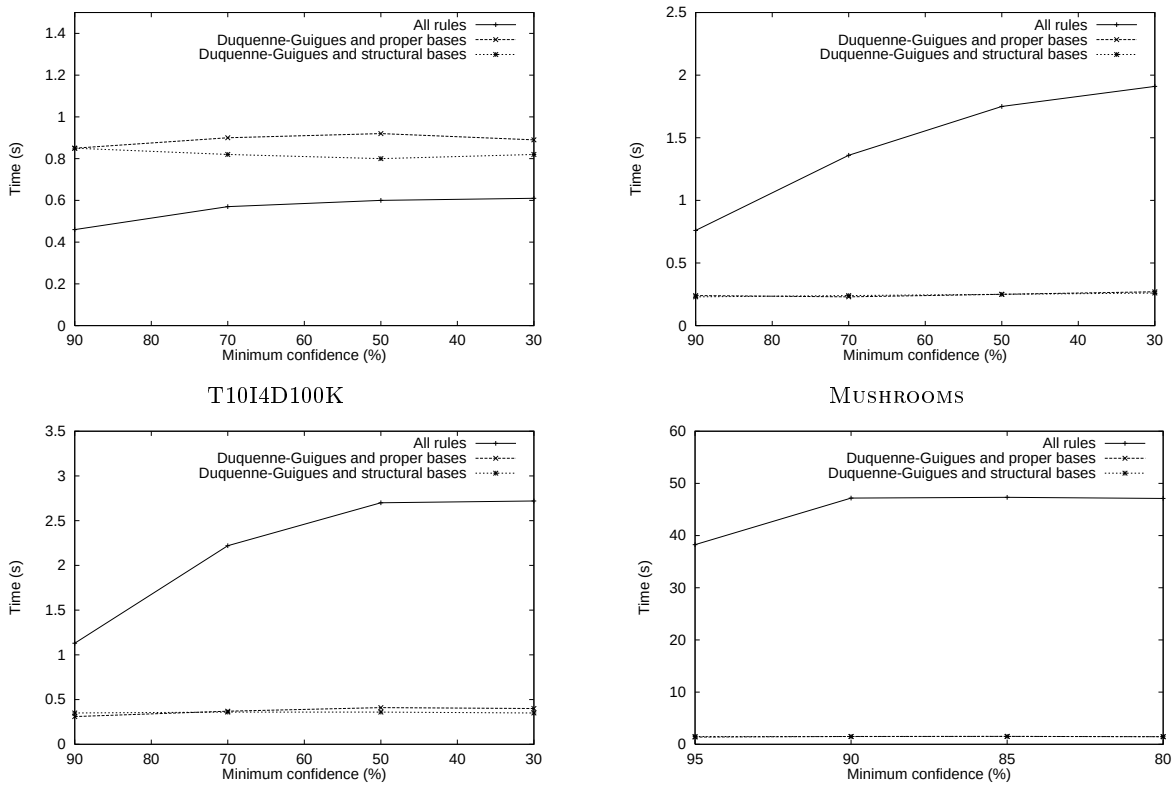


Figure 6: Execution Times of the Association Rule Derivation.

maximal consequent [10, 13]. Furthermore, the proper and structural bases for approximate rules are also smaller than the basis for approximate dependencies defined in [15]. Adapting our algorithms to the discovery of functional and approximate dependencies is an ongoing research.

Acknowledgements

The authors would like to gratefully acknowledge Rosine Cicchetti and Mohand-Said Hacid for their constructive comments.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *Proc. of the ACM SIGMOD Conference*, pages 207–216, May 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. of the 20th VLDB Conference*, pages 478–499, June 1994. Expanded version in IBM Research Report RJ9839.

[3] E. Baralis and G. Psaila. Designing templates for mining association rules. *Journal of Intelligent Information Systems*, 9(1):7–32, July 1997.

[4] R. J. Bayardo. Efficiently mining long patterns from databases. *Proc. of the ACM SIGMOD Conference*, pages 85–93, June 1998.

[5] R. J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Proc. of the 15th ICDE Conference*, pages 188–197, March 1999.

[6] G. Birkhoff. Lattices theory. In *Colloquium Publications XXV*. American Mathematical Society, 1967. Third edition.

[7] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlation. *Proc. of the ACM SIGMOD Conference*, pages 265–276, May 1997.

[8] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. *Proc. of the ACM SIGMOD Conference*, pages 255–264, May 1997.

[9] P. Burmeister. Formal concept analysis with ConImp: Introduction to the basic features. Technical report, Technische Hochschule Darmstadt, Germany, 1998.

[10] J. Demetrovics, L. Libkin, and I. B. Muchnik. Functional dependencies in relational databases: A lattice point of view. *Discrete Applied Mathematics*, 40:155–185, 1992.

[11] V. Duquenne and J.-L. Guigues. Famille minimale d'implication informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 24(95):5–18, 1986.

[12] B. Ganter and K. Reuter. Finding all closed sets: A general approach. In *Order*, pages 283–290. Kluwer Academic Publishers, 1991.

[13] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1998.

[14] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. *Proc. of the 21st VLDB Conference*, pages 420–431, September 1995.

[15] Y. Huhtala, J. Kärkkänen, P. Porkka, and H. Toivonen. Efficient discovery of functional and approximate dependencies using partitions. *Proc. of the 14th ICDE Conference*, pages 392–401, Febuary 1998.

[16] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. *Proc. of the 3rd CIKM Conference*, pages 401–407, November 1994.

[17] D. Lin and Z. M. Kedem. Pincer-search: A new algorithm for discovering the maximum frequent set. *Proc. of the 6th EDBT Conference*, pages 105–119, March 1998.

[18] B. Liu, W. Hsu, and S. Chen. Using general impressions to analyse discovered classification rules. *Proc. of the 3rd KDD Conference*, pages 31–36, August 1997.

[19] M. Luxenburger. Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines*, 29(113):35–55, 1991.

[20] H. Mannila and K. J. Räihä. Algorithms for inferring functional dependencies from relations. *Data & Knowledge Engineering*, 12(1):83–99, Febuary 1994.

[21] R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. *Proc. of the 22nd VLDB Conference*, pages 122–133, September 1996.

[22] R. T. Ng, V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained association rules. *Proc. of the ACM SIGMOD Conference*, pages 13–24, June 1998.

[23] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. *Proc. of the 7th ICDT Conference*, pages 398–416, January 1999.

[24] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.

[25] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases*, pages 229–248, 1991.

[26] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in larges databases. *Proc. of the 21st VLDB Conference*, pages 432–444, September 1995.

[27] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, December 1996.

[28] R. Srikant and R. Agrawal. Mining generalized association rules. *Proc. of the 21st VLDB Conference*, pages 407–419, September 1995.

[29] R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. *Proc. of the 3rd KDD Conference*, pages 67–73, August 1997.

[30] H. Toivonen. Sampling large databases for association rules. *Proc. of the 22nd VLDB Conference*, pages 134–145, September 1996.

[31] R. Wille. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23:493–515, 1992.

[32] M. J. Zaki and M. Ogihara. Theoretical foundations of association rules. *3rd SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, June 1998.

[33] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. *Proc. of the 3rd KDD Conference*, pages 283–286, August 1997.